PYTHON

Mode of Delivery:
-

- ILT Training across location - Bangalore,Chennai,Hyderabad, Pune,Delhi.
- All trainings will be delivered using our LMS.
- LMS access will be given to all learners, which will include a complete learning path, assessments, attendance tracking, feedback, reference material, and a participation certificate.

Assessments Model-

- Pre-Assessment- Just to check the knowledge of the participants.-
- Mid Assessment/Module Assessment (As required)
- Post-Assessment / Code Base Assessment.

Batch Size - 20-25 per batch.

Duration – 24 Hours

Prerequisite Skills: Basic Programming knowledge

Program Overview: This Python Course will help you to become a master in Python programming concepts such as Operators, Control statements, Loops, Data Types(Mutable and Immutable), List and dictionary comprehension, lambda, map, filter, reduce, definitions, database operations, regular expressions, data & file operations in Python, object-oriented concepts & various Python libraries used for ML such as Pandas, Numpy, Matplotlib (Visualization) and so on. This Python course is also a gateway towards your Data Science career

Skills covered:

- Python – An Introduction
- Python Features
- Flavors of Python
- Python v3.8, Anaconda and Pycharm Installation
- Python Operators
  - -Arithmetic Operators
  - -Assignment Operator
  - -Comparison Operator
  - -Logical Operator
  - -Bitwise Operator
  - -identity Operator
  - -Membership Operator

- Control statements
  - If condition
  - If..else
  - If_elif_else
  - Nested conditions
- Loops
  - for loop
  - while loop
  - Nested Loop
  - Recursive
  - Loops with else condition
  - pass , break and continue
- dir(), help()
- input()
- range()
- Data Structure
  - Immutable
    - Numbers
    - Strings
    - Tuple
  - Mutable
    - List
    - Dictionary
    - Set

- String Formation – Advanced formatting

- List Vs Tuple
- List Comprehension
- Deep Copy Vs Shallow Copy
- Dictionary Comprehension
- Decorators
- Generators
  - Iterator
  - yield statement
- Multithreading
  - run()
  - start()
  - join()
  - isAlive()
  - getter and setter

- Multiprocessing
- Multithread Vs Multiprocessing

- File operations

- 'with' statement
- modes
- seek
- tell
- read(r)
- write(w)
- append(q)
- read-write(r+, w+, a+)
- read-write
- Image reading and writing (rb,wb,ab)

- Modules/Packages

    - "os" and subprocess module
    - "sys" module
    - "math" module
    - "datetime" module
    - "time" module
    - "csv" module
    - Package creation and usage

- Definitions
    - Create a definitions
    - Handling the arguments
    - return statement
    - Various form of function arguments
    - doc string
    - Local and Global variable
    -
- lambda
- map
- filter
- reduce

- Memory management in Python
- Garbage collector in Python
- Regular Expressions
    - search
    - match
    - findall
    - split
    - sub

- Exception Handler
    - Common Exception Errors
    - Exception Hierarchy
    - Try
    - except
    - else
    - finally
    - Multiple exception handling
    - Raise exception

- XML and JSON Operation
    - XML process
    - JSON Process
    - JSON conversions

- Object Oriented Programming
    - Class
    - Objects
    - Constructor, destructor
    - Operator overloading
    - Methods and diff types of methods
    - Bound and unbound method
    - Decorator methods
    - Abstraction
    - Inheritance
    - Polymorphism
    - Encapsulation
    - Access Control – Public, Protected and Private

- Data Science libraries
    - Numpy
    - Pandas
    - Matplotlib/Scipy

REST API standards

1. Stateless Operations
2. Client-Server Architecture
3. Cacheable Responses
4. Layered System Design
5. Code on Demand
6. Uniform Interface

-- Connecting with APIs using libraries like request

1. Understanding HTTP Requests
2. Making Your First Request
3. Understanding Status Codes
4. Understanding Headers
5. Understanding Response
6. Using the Translate API
7. Handling Translate API Error Cases

-- Python frameworks like Flask

1. Installing Flask
2. Creating a Base Application
3. Using HTML templates
4. Setting up the Database
5. Displaying All Content
6. Displaying a filtered content
7. -- Session management and cookies
8. Setting Cookies in Flask
9. Getting Cookies in Flask
10. Login Application in Flask using cookies
11. Visitors counted through cookies
12. Client-side - sessions
13. Server-side - sessions
14. Session Example in Flask

-- Templating engine like jinja

1. Get Started With Jinja
2. Control the Flow in Jinja
3. Use Jinja With Flask

-- HTML/CSS/JS basics (Non python)

Basics of topics would be covered

Lab Set Up-

For the lab we need a Windows 10/ Ubuntu 20 desktop with the below configuration.
RAM: 8GB/16GB
Storage: 50 GB
core: 4
Internet open

Note: For lab machine participants must have installation permission.